Mitigating aggregated traffic of DHCP discover messages
draft-yang-dhc-ipv4-dis-01

Abstract

This document defines a new option DIS_MAX_RT which can mitigate aggregated traffic caused by discover messages the clients send to the server.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 13, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.
Table of Contents

1. Introduction ............................................. 3
2. Requirements Language ................................. 3
3. Potential Problems ................................... 3
4. DIS_MAX_RT and DIS_MAX_RT_OPTION ..................... 6
5. Security Considerations ............................... 7
6. IANA Considerations .................................. 7
7. References ............................................. 7
8. Acknowledgement ....................................... 7
Authors’ Addresses ...................................... 7
1. Introduction

In [RFC2131], there are no specific definitions for client’s operation if the server does not respond for the discover messages. In some cases, this will lead to an unacceptably high volume of aggregated traffic at a DHCPv4 server.

In [RFC3315], SOL_MAX_RT is defined as an option of DHCPv6 message to prevent the frequently requesting of clients, which reduce the aggregated traffic. In DHCPv4, there are no corresponding IPv4 options. Although the format of DHCPv4 is different with DHCPv6, it is also necessary to introduce similar option in DHCPv4 to keep the consistency between DHCPv4 and DHCPv6.

This document updates [RFC2131] by defining a new option DIS_MAX_RT which makes the DHCPv4 server mitigating aggregated traffic of client’s discover messages.

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Potential Problems

[RFC2131] defines the interaction between the DHCP server and clients. There are no specific description for client’s operation when the client does not receive the DHCPOFFER in response to its DHCPDISCOVER message. In normal IPv4 environment, clients will flood DHCPDISCOVER messages only when the server or link is broken. But in Dual-Stack scenarios, the problem becomes more frequent and serious. In IPv6 LAN/WLAN network or intranet, the core router or AC often plays the role of DHCP server, and the clients are several thousands of PC or mobile phones. If the server is configured in IPv6-only, the clients in dual-stack or IPv4-only, they will broadcast DHCPDISCOVER messages endlessly in the LAN or WLAN. The thousands clients will cause a DDOS-like attack to all the servers in the intranet.
To avoid this problem, most of the terminals create backoff algorithms which can help them retransmit DHCPDISCOVER message in different frequency according to their state machine in different Operating Systems, because there is no specific definition in RFCs to restrict the terminals behaviors when the server is down or in a dual-stack scenario as described upwards. But the same point of almost all the various Operating Systems is that they could not stop DHCPDISCOVER requests even to an IPv6-only server. We test some of the most popular terminals’ OS in WLAN, the results are illuminated as below.
DHCP Discovery Packages Time Table

<table>
<thead>
<tr>
<th>No</th>
<th>Windows7</th>
<th>Windows XP</th>
<th>IOS_5.0.1</th>
<th>Android_2.3.7</th>
<th>Symbian_S60</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0.1</td>
<td>7.8</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>3.9</td>
<td>0.1</td>
<td>1.4</td>
<td>10.3</td>
<td>2.5</td>
</tr>
<tr>
<td>3</td>
<td>13.3</td>
<td>4.1</td>
<td>3.8</td>
<td>2.4</td>
<td>17.9</td>
</tr>
<tr>
<td>4</td>
<td>30.5</td>
<td>12.1</td>
<td>7.9</td>
<td>4.1</td>
<td>33.9</td>
</tr>
<tr>
<td>5</td>
<td>62.8</td>
<td>29.1</td>
<td>16.3</td>
<td>8.4</td>
<td>36.5</td>
</tr>
<tr>
<td>6</td>
<td>65.9</td>
<td>35.8</td>
<td>24.9</td>
<td>8.6</td>
<td>reconnect</td>
</tr>
<tr>
<td>7</td>
<td>74.9</td>
<td>8.4</td>
<td>33.4</td>
<td>8.5</td>
<td>56.6</td>
</tr>
<tr>
<td>8</td>
<td>92.1</td>
<td>42.2</td>
<td>8.6</td>
<td>60.2</td>
<td>3.6</td>
</tr>
<tr>
<td>9</td>
<td>395.2</td>
<td>50.8</td>
<td>8.3</td>
<td>164</td>
<td>26</td>
</tr>
<tr>
<td>10</td>
<td>399.1</td>
<td>59.1</td>
<td>8.3</td>
<td>164</td>
<td>26</td>
</tr>
<tr>
<td>11</td>
<td>407.1</td>
<td>127.3</td>
<td>68.2</td>
<td>86.7</td>
<td>1.9</td>
</tr>
<tr>
<td>12</td>
<td>423.4</td>
<td>128.9</td>
<td>reconnect</td>
<td>32.1</td>
<td>2</td>
</tr>
<tr>
<td>13</td>
<td>455.4</td>
<td>131.1</td>
<td>2.2</td>
<td>106.7</td>
<td>20</td>
</tr>
<tr>
<td>14</td>
<td>460.4</td>
<td>135.1</td>
<td>4</td>
<td>111.4</td>
<td>4.7</td>
</tr>
<tr>
<td>15</td>
<td>467.4</td>
<td>143.4</td>
<td>8.3</td>
<td>120.6</td>
<td>9.2</td>
</tr>
<tr>
<td>16</td>
<td>483.4</td>
<td>151.7</td>
<td>8.3</td>
<td>134.9</td>
<td>14.3</td>
</tr>
<tr>
<td>17</td>
<td>842.9</td>
<td>160.4</td>
<td>8.7</td>
<td>136.8</td>
<td>1.9</td>
</tr>
<tr>
<td>18</td>
<td>846.9</td>
<td>317</td>
<td>168.8</td>
<td>8.4</td>
<td>reconnect</td>
</tr>
</tbody>
</table>

Terminals DHCPDISCOVER requests when Server’s DHCP module is down

Figure 2. Terminals DHCPDISCOVER requests when Server’s DHCP module is down For Windows7, it seems to initiate 8 times DHCPDISCOVER requests in about 300s interval. For WindowsXP, firstly it launches 9 times DHCPDISCOVER messages, but after that it cannot get any response from the server, then it initiates 5 times requests in one cycle in around 330s intervals, and never stop. For IOS5.0.1, it seems like WindowsXP. There are 10 times attempts in one cycle, and the interval is about 68s. Symbian_S60 uses the simplest backoff method, it launches DISCOVER in every 2 or 4 seconds. Android2.3.7 is the only Operating System which can stop DISCOVER request by disconnect its wireless connection. It reboot wireless and dhcp connection every 20 seconds. Obviously, DHCP server needs to weaken the traffic which is like DDoS attack caused by the clients when many DHCPv4 clients send discovery messages incessantly when the DHCPv4 server is configured no respond to discover messages or the IPv4 address pool is empty.
4. DIS_MAX_RT and DIS_MAX_RT_OPTION

In our experiments described upwards, some of the most popular OS will send several discover messages every 1 or 5 minutes, and send the message endlessly. So the DHCP server needs a mechanism to weaken the traffic.

It is necessary to define an uniform identification named DIS_MAX_RT for client to follow when it needs to retransmit DHCPDISCOVER. Client should retransmits the message in a period refer to the DIS_MAX_RT value. This parameter can be initiated by client and configured by DHCP server. Client must support this new option, and should deploy some backoff algorithm to avoid launch DISCOVER more frequently. Server must also support this option, and could refill the parameter according to its state.

According to the definition of DHCP option in RFC2132, a new option named DIS_MAX_RT_OPTION is defined. The format of DIS_MAX_RT_OPTION is:

```
+--------+--------+--------+--------+--------+--------+
|Code    |Len     |   T1   |   T2   |   T3   |   T4   |
+--------+--------+--------+--------+--------+--------+
```

Code DIS_MAX_RT_OPTION (TBD).
Len 4.
T1-T4 4 octets, Overriding value for DIS_MAX_RT in seconds.

DIS_MAX_RT_OPTION

The DIS_MAX_RT_OPTION option needs IANA to assign a new Code to indicate and its length (Len value) is 4 octets. From T1 to T4, there are 4 octets space to indicate the max retransmission time period. MRT(T1-T4) identifies the interval time client sends two concatenated DISCOVER message. MRT must > 0; When MRT=FFFF, client should not send DISCOVER any more. A DHCPv4 client MUST include the DIS_MAX_RT_OPTION in any message it sends. The DHCPv4 server MAY include the DIS_MAX_RT_OPTION code in any response it sends to a client that has included the DIS_MAX_RT option code in a request message. The process of this option is described below: 1. Client must initial the time parameter by any random algorithm or any others, and set T1-T4 in DIS_MAX_RT_OPTION. IF client receives DIS_MAX_RT_OPTION from server, it should retransmit DISCOVER according the MRT in the option. As a result of receiving this
option, the DHCPv4 client MUST NOT send any request messages more frequently than allowed by the retransmission mechanism defined by their OS. Client should deploy backoff algorithm to retransmit the message if it does not receive any message from server until the backoff time is triggered. 2. When server receives a request including a DIS_MAX_RT_OPTION, it MAY ignore the value of DIS_MAX_RT and assign a new value in the response to make the client refresh its DIS_MAX_RT. It can change MRT longer than the initialized time if the IPv4 address pool is empty or according to the administrator’s configuration. Server can also change the value to FFFF if it does not want to support any more IPv4 address request or in a normal address allocation process in DHCPOFFER or any other messages.

5. Security Considerations

The security problem is under discussion.

6. IANA Considerations

IANA is requested to assign an option code from the "DHCP Option Codes" Registry for OPTION_DIS_MAX_RT.

7. References

(1) RFC[2131] Dynamic Host Configuration Protocol
(2) RFC[2132] DHCP Options and BOOTP Vendor Extensions
(3) RFC[3315] Dynamic Host Configuration Protocol for IPv6(DHCPv6)
(4) "draft-droms-dhc-dhcpv6-solmaxrt-update-02" Modification to Default Value of SOL_MAX_RT

8. Acknowledgement

Thanks for the authors of "draft-droms-dhc-dhcpv6-solmaxrt-update-02" and the contributions from Lianyuan Li.
Authors’ Addresses

Tianle Yang  
China Mobile  
32, Xuanwumenxi Ave.  
Xicheng District, Beijing 01719  
China  
Email: yangtianle@chinamobile.com

Li Lianyuan  
China Mobile  
32, Xuanwumenxi Ave.  
Xicheng District, Beijing 01719  
China  
Email: lilianyuan@chinamobile.com

Qiongfang Ma  
China Mobile  
32, Xuanwumenxi Ave.  
Xicheng District, Beijing 01719  
China  
Email: maqiongfang@chinamobile.com